Forschungsschwerpunkt
Algorithmen und mathematische Modellierung

Graz University of Technology

Up- and downgrading the 1-median in a network

Elisabeth Gassner

Project Area(s):
Effizient lösbare kombinatorische Optimierungsprobleme

Institut für Optimierung und Diskrete Mathematik (Math B)

Report 2007-16, November 2007

Up- and downgrading the 1-median in a network*

Elisabeth Gassner

Technische Universität Graz, Institut für Mathematik B, Steyrergasse 30, 8010 Graz, Austria, gassner@opt.math.tu-graz.ac.at

Abstract. While classical location problems deal with finding optimal locations for facilities, the task of the corresponding upgrading (downgrading) version is to change the underlying network within certain bounds such that the optimal objective value that can be obtained in the modified network is as good (bad) as possible. In this paper we allow to change the vertex weights within given bounds such that a linear budget constraint is satisfied. For the upgrading 1-median problem an $\mathcal{O}(n^2)$ time algorithm is suggested. The downgrading 1-median problem is shown to be solvable in polynomial time. For the special case of a tree a concavity property leads to an $\mathcal{O}(n \log n)$ time algorithm.

1 Introduction

In this paper we consider a network up- and downgrading problem where the goal is to change vertex weights within certain limits such that the quality of the resulting optimal facility location is maximized or minimized.

Assume that a company plans to open a new factory with storage capacity B. Moreover, assume that warehouses with storage capacity w_i for $i = 1, \ldots, n$ already exist and the total demand of raw materials in the factory is $\sum_{i=1}^{n} w_i$. Since the factory can store materials (for which the transportation cost is equal to 0) the company wants to fix the quantity stored in each warehouse as well as an optimal location for the factory in order to minimize the total transportation cost from the warehouses to the factory.

This application can be modeled by the upgrading 1-median problem where the task is to change the vertex weights (here storage capacity) within certain limits (the factory takes at most B units) such that the total transportation cost for an optimal location of the factory is minimized.

Observe that we do not fix a location and maximally improve its quality by changing the vertex weights but we change the vertex weights in a first step and in a second step an optimal location with respect to the new weights is determined.

Up- and downgrading problems are special network modification problems. In the upgrading 1-median we have to change the parameters of the network

* This research has been supported by the Austrian Science Fund (FWF) Project P18918-N18.

within certain limits such that the optimal objective value in respect to the modified parameters is minimized. For the downgrading version the task is to change the parameters in order to maximize the optimal objective value in the modified network. Both, the up- and downgrading version are applied to the 1-median problem which belongs to the most important basic models of location problems. For an introduction to location problems the reader is referred to Kariv and Hakimi [18], Mirchandani and Francis [21] and Drezner and Hamacher [9]).

1.1 Related Problems

A general variant of up- and downgrading problems is of the following form: Given an (minimization) optimization problem with optimal objective value $z(w)$ where w is a vector of parameters. The task of the corresponding upgrading problem is to modify the parameters within certain bounds such that $z(\tilde{w})$ is minimized. The downgrading version is to maximize $z(\tilde{w})$. This concept was applied to several classical combinatorial optimization problems, e. g., shortest and longest path (Fulkerson and Harding [11] and Hambrusch and Tu [16]), network flows (Phillips [22]), minimum spanning trees and Steiner trees (Frederickson and Solis-Oba [10], Drangmeister et al. [8] and Krumke et al. [19]). A general framework for up- and downgrading versions of 0/1-combinatorial optimization problems was investigated by Burkard, Klinz and Zhang [4] and Burkard, Lin and Zhang [5]. Moreover, up- and downgrading versions for the 1-center problem on networks were investigated (Gassner [13]).

Upgrading problems are closely related to reverse problems where a feasible solution is given and the task is to modify parameters within certain limits in order to maximally improve the objective value of the given feasible solution. In case of location problems the reader is referred to Burkard, Gassner and Hatzl [2, 3] and Zhang, Yang and Cai [23, 24].

Moreover, inverse problems are also network modification problems where a feasible solution is given. But here the task is to modify parameters within certain bounds and at minimum cost such that the given solution becomes optimal with respect to the modified parameters. Inverse location problems were investigated in [6, 7, 12].

1.2 Problem definition and organization of the paper

This paper is dedicated to up- and downgrading versions of the 1-median problem.

The 1-median problem is to locate a facility in a network such that the sum of weighted shortest distances from the vertices to the facility is minimized, i. e., given a graph $G = (V, E)$ with edge lengths $\ell_e \in R_+$ for $e \in E$ and vertex weights $w_v \in \mathbb{R}_+$ for $v \in V$ the task is to find a vertex $x \in V$ in the graph which minimizes

$$f(x) = \sum_{v \in V} w_v d(v, x)$$

where $d(v, x)$ denotes the shortest distance in G from v to x.

The 1-median problem is a generalization of the p-median problem where the task is to find a subset $X \subseteq V$ of p vertices such that

$$f(X) = \sum_{v \in V} w_v \min_{x \in X} d(v, x)$$

is minimized.

Instead of restricting the potential set of locations to the set of vertices, one may consider the problem where it is allowed to place the facilities on every point on the graph, i. e., the set of vertices and all points in the interior of edges. The corresponding problem is called absolute p-median problem. However, if all vertex weights w_v are nonnegative then the absolute p-median and the p-median problem coincide. Hakimi [15] proved the so-called vertex-optimality property of the p-median problem: Given an instance of the absolute p-median problem with nonnegative vertex weights, then there exists an absolute p-median consisting only of vertices. This property immediately leads to a polynomial time algorithm for the p-median if p is fixed since there exists an optimal solution among $\binom{n}{p}$ alternatives. Moreover, the objective value of each subset can be determined in polynomial time. Hence, the 1-median problem can be solved in polynomial time by determining the objective value of every vertex and finally taking the best one. If p is part of the input then the p-median problem is in general $\mathcal{NP}$-hard (Kariv and Hakimi [18]). However, for the special case on trees the problem is again solvable in polynomial time. Kariv and Hakimi suggested a $\mathcal{O}(n^2p^2)$ time algorithm for the p-median problem on trees. The 1-median problem on a tree can even be solved in linear time due the convexity of the objective value along a path (Goldman [14]).

In this paper we consider a variant of the 1-median problem where the network is modified before the facility is located on the network. Up- and downgrading problems can be seen as bilevel problems where one decision maker (the actor) changes the vertex weights within certain limits and another decision maker (the location planner or reactor) locates the facility optimally with respect to the new vertex weights. The goal of the location planner is to minimize the 1-median objective value. Depending on whether the goals of actor and reactor are the same or are conflicting we speak about up- or downgrading problems. In the upgrading version the actor wants to minimize the 1-median objective value (i. e., actor and reactor have the same goal) while in the downgrading version the actor seeks to maximize the 1-median objective value while the location planner wants to minimize this value. Hence, the downgrading version is a max-min-problem.

Denote the optimal 1-median objective value with respect to vertex weights w by $z(w)$. Then the task of the upgrading 1-median problem is to increase the weights by $\delta = (\delta_v)_{v \in V}$ such that δ is a feasible vertex weight modification and $z(w - \delta)$ is minimized. Analogously, the downgrading 1-median problem is to find a feasible weight modification δ such that $z(w + \delta)$ is maximized.

A vertex weight modification $\delta = (\delta_v)_{v \in V}$ is called feasible if a budget constraint is met and the modifications are within certain bounds: Let $c_v \in \mathbb{R}_+$ for

$v \in V$ denote the cost of changing the weight of vertex v by one unit and let $u_v \in \mathbb{R}_+$ for $v \in V$ be an upper bound for the modification of the weight of vertex v. Moreover, we are given a total budget B. Then δ is feasible if $\delta \in \Delta$ with

$$\Delta = \left\{ \delta \,\middle|\, \sum_{v \in V} c_v \delta_v \leq B \text{ and } 0 \leq \delta_v \leq u_v \text{ for all } v \in V \right\}.$$

Hence, we can define the upgrading and downgrading problems: Let $G = (V, E)$ be a graph with vertex weights $w_v \in \mathbb{R}_+$, cost coefficients $c_v \in R_+$ and bounds $u_v \in \mathbb{R}_+$ for all $v \in V$, edge lengths $\ell_e \in \mathbb{R}_+$ for all $e \in E$ and a total budget B.

Then the upgrading 1-median problem, Up1Median for short, is to solve

$$\min_{\delta \in \Delta} z(w - \delta) = \min_{\delta \in \Delta} \min_{x \in V} \sum_{v \in V} (w_v - \delta_v) d(v, x).$$

And the downgrading 1-median problem, Down1Median for short, is to solve

$$\max_{\delta \in \Delta} z(w + \delta) = \max_{\delta \in \Delta} \min_{x \in V} \sum_{v \in V} (w_v + \delta_v) d(v, x).$$

In this paper we will present an $\mathcal{O}(n^2)$ time algorithm for Up1Median provided that the distance matrix is given. For Down1Median a linear programming formulation is given. Hence, Down1Median can be solved in polynomial time. If the underlying graph is a tree, a type of concavity property can be shown for Down1Median which leads to an $\mathcal{O}(n \log n)$ time algorithm.

1.3 Notation

Throughout this paper we will use the following notation: Let $G = (V, E)$ be a graph. Then $n = |V|$ is the number of vertices and $m = |E|$ is the number of edges.

Let $G = (V, E)$ be a graph with vertex weights $w_v \in \mathbb{R}_+$ and let $X \subseteq V$ be a subset of vertices. Then $w(X) = \sum_{v \in X} w_v$ denotes the total weight of vertices in X. Let $\delta_v \in \mathbb{R}_+$ be additional vertex weights then $(w + \delta)(X) = \sum_{v \in X} (w_v + \delta_v)$. If no ambiguity is possible then we write $w(H)$ instead of $w(\tilde{V})$ for $H = (\tilde{V}, \tilde{E})$ is a subgraph of G.

Let $x \in V$ then the neighbourhood of x is denoted by $\Gamma(x) = \{y \in V \mid (x, y) \in E\}$.

Let $T = (V, E)$ be a tree and $v \in V$ then $\mathcal{T}(v)$ denotes the set of subtrees of T that arise if vertex v is deleted. Let $T' = (V', E')$ and $T'' = (V'', E'')$ be two vertex disjoint subtrees of T then $T' + T'' = (V' \cup V'', E' \cup E'')$ denotes the union of both subtrees which is again a subtree of T. And finally, $T - T' = (V - V', \{(i, j) \in E \mid i, j \in V \setminus V'\})$ is a subgraph of T which is induced by the vertex set $V \setminus V'$.

2 Upgrading the 1-median

This section is dedicated to upgrading the 1-median in a graph. Given a graph $G = (V, E)$, vertex weights $w_v \in \mathbb{R}_+$, cost coefficients $c_v \in \mathbb{R}_+$ and bounds $u_v \leq w_v \in \mathbb{R}_+$ for all $v \in V$, edge lengths $\ell_e \in \mathbb{R}_+$ for all edges $e \in E$ and a budget B, the task is to solve

$$\min_{\delta \in \Delta} \min_{x \in V} \sum_{v \in V} (w_v - \delta_v) d(v, x).$$

Since we can interchange the first two minimum-operations, Down1Median is equivalent to

$$\min_{x \in V} \min_{\delta \in \Delta} \sum_{v \in V} (w_v - \delta_v) d(v, x)$$

which leads to n reverse 1-median problems. For each vertex $x \in V$ the corresponding 1-median objective value has to be maximally improved by a feasible weight modification. Finally, the best upgraded objective value is equal to the optimal objective value of Up1Median. Each reverse 1-median problem is a continuous knapsack problem which can be solved in $\mathcal{O}(n)$ time (Balas and Zemel [1]). Hence, Up1Median can be solved by solving n continuous knapsack problems and finally comparing the obtained maximally improved objective values.

Theorem 1. *Upgrading the* 1*-median by vertex weight modifications can be solved in* $\mathcal{O}(n^2)$ *time provided that the distance matrix is given.*

We conclude this section with an example which shows some properties of Up1Median.

In general the 1-median changes after an optimal weight modification. Consider the following instance of Up1Median given in Figure 1 with $B = 2$.

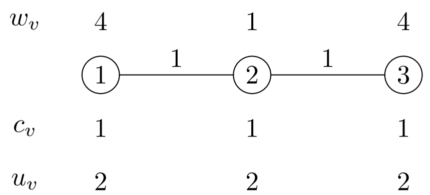

Fig. 1. Instance of Up1Median.

The objective values with respect to the original weights are $f(1) = f(3) = 9$ and $f(2) = 8$. Hence, vertex 2 is the unique 1-median. Now fix vertex 1 and maximally improve its objective value. Then $\delta_1 = \delta_2 = 0$ and $\delta_3 = 2$ is an optimal improvement and hence the new objective value of vertex 1 is equal to $\tilde{f}(1) = 5$. Due to symmetry $\delta_1 = 2$ and $\delta_2 = \delta_3 = 0$ is an optimal improvement of vertex 3 with improved objective value $\tilde{f}(3) = 5$. If vertex 2 is maximally improved then $\delta_1 = \delta_3 = 1$ and $\delta_2 = 0$ and hence the improved objective value of vertex 2 is $\tilde{f}(2) = 6$. Hence, either vertex 1 or vertex 3 are 1-medians after an optimal weight modification and hence vertex 2 looses its optimality.

Observe, that although the objective values for fixed weights are convex along a path, the maximally improved objective values do not have to property any more.

3 Downgrading the 1-median on general graphs

In this section we investigate the problem of downgrading the 1-median on graphs. Down1Median is shown to be solvable in polynomial time since it can be written as linear programming problem.

Let us recall the definition of Down1Median: Given a graph $G = (V, E)$ with vertex weights $w_v \in \mathbb{R}_+$, cost coefficients $c_v \in \mathbb{R}_+$ and bounds $u_v \in \mathbb{R}_+$ for all vertices $v \in V$, edge lengths $\ell_e \in \mathbb{R}_+$ for $e \in E$ and a total budget B. Down1Median is to

$$\max_{\delta \in \Delta} \min_{x \in V} \sum_{v \in V} (w_v + \delta_v) d(v, x).$$

Observe that maximally downgrading the objective value of each vertex and finally taking the best solution does not lead to an optimal solution of Down1Center in general. Consider the following example with $G = (V, E)$ as given in Figure 2.

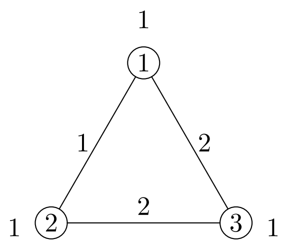

Fig. 2. Instance of Down1Median with $w_v = c_v = 1$ and $u_v = 2$ for all $v \in V$ and $B = 2$.

Before any weight modification we have $f(1) = f(2) = 3$ and $f(3) = 4$, i. e., vertex 1 and vertex 2 are 1-medians. Now fix vertex 1 and maximally degrade the objective value of vertex 1. Then $\delta_1 = \delta_2 = 0$ and $\delta_3 = 2$ and the new objective values are $\tilde{f}(1) = \tilde{f}(3) = 7$ and $\tilde{f}(2) = 4$. Hence, the new optimal objective value is equal to 4. Due to symmetry we get the same result for vertex 3. If we fix vertex 2 then vertex 1 and vertex 3 have the same efficiency. Let $\delta_1 = \lambda$ and $\delta_2 = 2 - \lambda$ then $\tilde{f}(1) = 5 - \lambda$, $\tilde{f}(2) = 3 + \lambda$ and $\tilde{f}(3) = 8$. Hence, the 1-median objective value is maximal for $\lambda = 1$ and the corresponding optimal objective value is equal to 4. We conclude that this procedure yields the optimal objective value 4. Observe that if vertex $x \in V$ is fixed and its objective value is maximized, then in general vertex x is not a 1-median with respect to the new vertex weights. Moreover, the unique optimal solution of Down1Median is $\delta_1 = \delta_3 = \frac{3}{7}$ and $\delta_2 = \frac{8}{7}$ with objective value $\frac{40}{7}$ which is strictly greater than the maximum value we got if we fix each vertex separately. In addition this example shows that in general every weight - even the weight of the vertex which will be the 1-median after an optimal weight modification - has to be changed in an optimal solution.

However, Down1Median can be solved in polynomial time since it can be formulated by the following linear program:

$$\begin{aligned}
\max_{\delta, L} \quad & L \\
\text{s.t.} \quad & \sum_{v \in V} (w_v + \delta_v) d(v, x) \geq L && \text{for all } x \in V \\
& \sum_{v \in V} c_v \delta_v \leq B \\
& 0 \leq \delta_v \leq u_v && \text{for all } v \in V
\end{aligned}$$

Theorem 2. *The problem of downgrading the* 1*-median can be solved in polynomial time.*

4 Downgrading the 1-median in a tree

In this section we consider the problem of maximally downgrading the 1-median in a tree. As main theorem we show a concavity property which leads to an $\mathcal{O}(n \log n)$ time algorithm for Down1Median on trees.

Observe that there exists an optimality criterion for the 1-median problem on a tree which is independent of the edge lengths:

Theorem 3 (Goldman [14]). *Let $T = (V, E)$ be a tree, $w_v \in \mathbb{R}_+$ for $v \in V$ be vertex weights and $\ell_e \in \mathbb{R}_+$ for $e \in E$ be edge lengths. Then vertex $x \in V$ is a* 1*-median if and only if*

$$w(T') \leq \frac{1}{2} w(T) \qquad \text{for all} \quad T' \in \mathcal{T}(x).$$

Let $g(x)$ for $x \in V$ denote the maximal objective value of vertex x which can be obtained by modifying the weights by a feasible δ and simultaneously making sure that x is a 1-median with respect to the new weights. Using the above optimality criterion for the 1-median problem we can define $g(x)$ in the following way:

$$
\begin{aligned}
(\mathrm{P}(x)) \qquad g(x) = \max \; & \sum_{v\in V} (w_v + \delta_v) d(v,x) & & \\
\text{s.t.} \; & \sum_{v\in V} c_v \delta_v \le B & & (1)\\
& 0 \le \delta_v \le u_v & \text{for all } v \in V \quad & (2)\\
& (w+\delta)(T') \le \frac{1}{2}(w+\delta)(T) & \text{for all } T' \in \mathcal{T}(x) \quad & (3)
\end{aligned}
$$

Obviously, $\max_{x\in V} g(x)$ is equal to the optimal objective value of Down1Median on a tree. Before investigating $\mathrm{P}(x)$ we turn our attention to the corresponding relaxed problem $\mathrm{P}'(x)$:

$$
\begin{aligned}
(\mathrm{P}'(x)) \qquad \max \; & \sum_{v\in V} (w_v + \delta_v) d(v,x) & & \\
\text{s.t.} \; & \sum_{v\in V} c_v \delta_v \le B & & (4)\\
& 0 \le \delta_v \le u_v & \text{for all } v \in V \quad & (5)
\end{aligned}
$$

Observe that $\mathrm{P}'(x)$ is a continuous knapsack problem. Therefore, we define the efficiency of a vertex $v \in V$ with respect to x by

$$\mathrm{eff}_x(v) = \frac{d(v,x)}{c_v}.$$

Let us call a solution $\xi = (\xi_i)_{i\in V}$ to be an efficient solution for x if ξ is a feasible solution of $\mathrm{P}'(x)$, the budget constraint (1) is met with equality and whenever $\xi_i < u_i$ and $\xi_j > 0$ $(i \ne j)$ holds then $\mathrm{eff}_x(i) \le \mathrm{eff}_x(j)$. Clearly, an optimal solution of $\mathrm{P}'(x)$ is an efficient solution for x. Moreover, an optimal solution of $\mathrm{P}(x)$ such that all constraints of type (3) are met with strict inequality is optimal for $\mathrm{P}'(x)$ and therefore an efficient solution for x.

The goal of the following investigations is to compare the optimal objective values of two adjacent vertices. In order to do this, we first compare efficient solutions for adjacent vertices and then optimal solutions for $\mathrm{P}(x)$ and $\mathrm{P}'(x)$.

For the proofs of the following lemmata we will often make use of the following technical lemma:

Lemma 1. *Let I be a set of elements and $c_i > 0$ for $i \in I$. Assume that*

$$\sum_{i\in I} c_i \xi_i \ge \sum_{i\in I} c_i \eta_i \qquad (6)$$

$$\sum_{i\in I} \xi_i < \sum_{i\in I} \eta_i \qquad (7)$$

then there exist two elements j and k such that $\xi_j < \eta_j$ and $\xi_k > \eta_k$ and $c_j < c_k$ holds.

Proof. The existence of element j is guaranteed by (7). Assume that $\xi_i \leq \eta_i$ for all $i \in I$ and $\xi_j < \eta_j$. Then we get a contradiction to (6). Therefore, there exists an element k with $\xi_k > \eta_k$. It remains to show that $c_j < c_k$. Let

$$A = \{i \in I \mid \xi_i < \eta_i\}, \qquad B = \{i \in I \mid \xi_i > \eta_i\}$$

Both sets are known to be non-empty. Let

$$c_{\min} = \min_{i \in A} c_i, \qquad c_{\max} = \max_{i \in B} c_i$$

and assume that $c_{\min} \geq c_{\max}$ holds. Then,

$$\begin{aligned} 0 &\leq \sum_{i \in A} c_i(\xi_i - \eta_i) + \sum_{i \in B} c_i(\xi_i - \eta_i) \\ &\leq c_{\min} \sum_{i \in A} (\xi_i - \eta_i) + c_{\max} \sum_{i \in B} (\xi_i - \eta_i) \\ &\leq c_{\min} \left(\sum_{i \in A} (\xi_i - \eta_i) + \sum_{i \in B} (\xi_i - \eta_i) \right) \\ &= c_{\min} \left(\sum_{i \in I} \xi_i - \sum_{i \in I} \eta_i \right) < 0 \end{aligned}$$

which leads to a contradiction. Hence, $c_{\min} < c_{\max}$ and therefore there exist elements $j \in A$ and $k \in B$ with $c_j < c_k$.

The following lemma compares efficient solutions of adjacent vertices: Let $(x, y) \in E$ and let λ be an efficient solution for x. Moreover, let vertex y lie in subtree $T' \in \mathcal{T}(x)$. We prove that if the weight of T' with respect to λ is small then there exists an efficient solution $\tilde{\lambda}$ for y such that the weight of T' is also small. The intuition behind this lemma is, that if we move from x to y then the efficiencies of the elements in T' decrease while the remaining efficiencies increase. Therefore, $\tilde{\lambda}$ tends to take less elements out of T'. Formally, we prove the following result:

Lemma 2. *Let $(x, y) \in E$ and let λ be an efficient solution of x with*

$$(w + \lambda)(T') < \frac{1}{2}(w + \lambda)(T) \tag{8}$$

for $T' \in \mathcal{T}(x)$ with $y \in T'$. Then there exists an efficient solution $\tilde{\lambda}$ for y with

$$(w + \tilde{\lambda})(T') \leq \frac{1}{2}(w + \tilde{\lambda})(T).$$

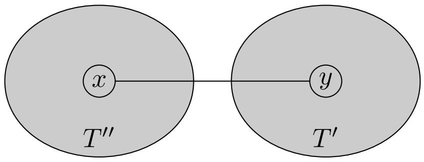

Fig. 3. Illustration for the proof of Lemma 2.

Proof. Let $T'' = T - T'$, i. e., $T'' \in \mathcal{T}(y)$ with $x \in T''$ (see Figure 3) and assume that every efficient solution $\tilde{\lambda}$ for y satisfies

$$(w + \tilde{\lambda})(T') > (w + \tilde{\lambda})(T'').$$

The proof is split into three subcases:

1. $\tilde{\lambda}_i \leq \lambda_i$ for all $i \in T'$:

   Then $(w+\tilde{\lambda})(T') \leq (w+\lambda)(T')$ holds. Together with (8) we get $(w+\tilde{\lambda})(T'') < (w + \lambda)(T'')$ and hence
   $$\sum_{j\in T''} \tilde{\lambda}_j < \sum_{j\in T''} \lambda_j.$$
   Since $\tilde{\lambda}$ is efficient and therefore uses the whole budget we get
   $$\sum_{j\in T''} c_j\tilde{\lambda}_j = B - \sum_{i\in T'} c_i\tilde{\lambda}_i \geq B - \sum_{i\in T'} c_i\lambda_i \geq \sum_{j\in T''} c_j\lambda_j.$$
   Lemma 1 implies that there exist two vertices $j, k \in T''$ with $\tilde{\lambda}_j < \lambda_j$ and $\tilde{\lambda}_k > \lambda_k$ and $c_j < c_k$. Observe that $\lambda_j > 0$ and $\lambda_k < u_k$. Since λ is an efficient solution for x, we have $\text{eff}_x(j) \geq \text{eff}_x(k)$ and
   $$\begin{aligned} \text{eff}_x(j) - \text{eff}_x(k) =& \text{eff}_y(j) - \frac{\ell(x,y)}{c_j} - \text{eff}_y(k) + \frac{\ell(x,y)}{c_k} \geq 0 \\ \text{eff}_y(j) - \text{eff}_y(k) \geq& \ \ell(x,y)\left(\frac{1}{c_j} - \frac{1}{c_k}\right) > 0 \end{aligned}$$
   Hence, $\text{eff}_y(j) > \text{eff}_y(k)$ and therefore the efficiency of $\tilde{\lambda}$ implies either $\tilde{\lambda}_j = u_j$ or $\tilde{\lambda}_k = 0$ which contradicts $\tilde{\lambda}_j < \lambda_j \leq u_j$ and $\tilde{\lambda}_k > \lambda_k \geq 0$.
2. Let $i \in T'$ with $\tilde{\lambda}_i > \lambda_i$ and assume that there exists a vertex $j \in T''$ with $\tilde{\lambda}_j < \lambda_j$:

   Then $\lambda_i < u_i$ and $\lambda_j > 0$ and therefore the efficiency of λ implies $\text{eff}_x(i) \leq \text{eff}_y(j)$. But
   $$\begin{aligned} \text{eff}_x(j) - \text{eff}_x(i) &= \text{eff}_y(j) - \frac{\ell(x,y)}{c_j} - \text{eff}_y(i) - \frac{\ell(x,y)}{c_i} \geq 0 \\ \text{eff}_y(j) - \text{eff}_y(i) &\geq \ell(x,y)\left(\frac{1}{c_j} + \frac{1}{c_k}\right) > 0 \end{aligned}$$

Again $\text{eff}_y(j) > \text{eff}_y(i)$ leads to a contradiction to the efficiency of $\tilde{\lambda}$.

3. Assume that $\tilde{\lambda}_j \geq \lambda_j$ for all $j \in T''$ and there exists a vertex $i \in T'$ with $\tilde{\lambda}_i > \lambda_i$:

   Observe that since λ is an efficient solution, it uses the whole budget. We use the same ideas as for the first case to show that

   $$\sum_{k \in T'} \tilde{\lambda}_k > \sum_{k \in T'} \lambda_k$$

   and

   $$\sum_{k \in T'} c_k \tilde{\lambda}_k = B - \sum_{j \in T''} c_j \tilde{\lambda}_j \leq B - \sum_{j \in T''} c_j \lambda_j = \sum_{i \in T'} c_i \lambda_i.$$

   Lemma 1 implies the existence of $j, k \in T'$ with $\tilde{\lambda}_j > \lambda_j$, $\tilde{\lambda}_k < \lambda_k$ and $c_k > c_j$. Due to the efficiency of λ we have $\text{eff}_x(j) \leq \text{eff}_x(k)$ and

   $$\begin{aligned} \text{eff}_x(k) - \text{eff}_x(j) &= \text{eff}_y(k) + \frac{\ell(x,y)}{c_k} - \text{eff}_y(j) - \frac{\ell(x,y)}{c_j} \geq 0 \\ \text{eff}_y(k) - \text{eff}_y(j) &\geq \ell(x,y)\left(\frac{1}{c_j} - \frac{1}{c_k}\right) > 0 \end{aligned}$$

   Again we get a contradiction to the efficiency of $\tilde{\lambda}$.

In a next step we are interested in the relationship between an optimal solution of $\mathrm{P}(x)$ and an optimal solution of $\mathrm{P}'(x)$. Since $\mathrm{P}'(x)$ is a continuous knapsack problem, we prefer solving $\mathrm{P}'(x)$ instead of $\mathrm{P}(x)$. If $\mathrm{P}'(x)$ has an optimal solution λ that is feasible for $\mathrm{P}(x)$ then λ is optimal for $\mathrm{P}(x)$. However, if λ is infeasible for $\mathrm{P}(x)$ then we have still enough information about an optimal solution $\mathrm{P}(x)$.

Lemma 3. *Let λ be an optimal solution of $P'(x)$ and $T' \in \mathcal{T}(x)$. If*

$$(w+\lambda)(T') \geq \frac{1}{2}(w+\lambda)(T)$$

holds then there exists an optimal solution δ of $P(x)$ with

$$(w+\delta)(T') = \frac{1}{2}(w+\delta)(T).$$

Proof. If $(w+\lambda)(T') = \frac{1}{2}(w+\lambda)(T)$ then $(w+\lambda)(\tilde{T}) \leq (w+\lambda)(T') = \frac{1}{2}(w+\lambda)(T)$ holds for all $\tilde{T} \in \mathcal{T}(x)$. Hence, λ is a feasible solution of $\mathrm{P}(x)$ and therefore optimal and has the required equality-property.

Assume that $(w+\lambda)(T') > \frac{1}{2}(w+\lambda)(T)$ and $(w+\delta)(T') < \frac{1}{2}(w+\delta)(T)$ for all optimal solutions δ of $\mathrm{P}(x)$. We distinguish two types of solutions:

1. Type: There exists a subtree $T'' \in \mathcal{T}(x)$ with $(w+\delta)(T'') = \frac{1}{2}(w+\delta)(T)$.
2. Type: $(w+\delta)(T'') < \frac{1}{2}(w+\delta)(T)$ for all $T'' \in \mathcal{T}(x)$.

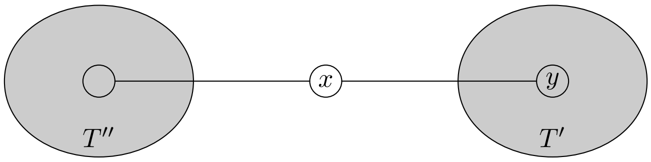

Fig. 4. Illustration for the proof of Lemma 3.

Assume that we are given a solution δ of the 1. Type (see Figure 4):

- If $\lambda_i \leq \delta_i$ for all $i \in T - T''$ then $(w+\lambda)(T-T'') \leq (w+\delta)(T-T'')$ and therefore

$$\begin{aligned}(w+\delta)(T'') &= (w+\delta)(T-T'') \geq (w+\lambda)(T-T'') \\ &> (w+\lambda)(T') > (w+\lambda)(T-T') > (w+\lambda)(T'').\end{aligned}$$

  On the other hand, we have

$$\sum_{i\in T''} c_i\lambda_i = B - \sum_{j\in T-T''} c_j\lambda_j \geq B - \sum_{j\in T-T''} c_j\delta_j \geq \sum_{i\in T''} c_i\delta_i.$$

  Due to Lemma 1 there are two vertices $j,k \in T''$ with $\delta_j > \lambda_j$ and $\delta_k < \lambda_k$ and $c_j < c_k$. The optimality of λ implies $\mathrm{eff}_x(j) \leq \mathrm{eff}_x(k)$. There exists an $\epsilon > 0$ such that the solution $\hat{\delta}$ obtained by δ where δ_j is decreased by $\frac{\epsilon}{c_j}$ and δ_k increased by $\frac{\epsilon}{c_k}$ satisfies the bound and budget constraints. Moreover, the weight of subtree T'' increases by $\varepsilon\left(\frac{1}{c_k} - \frac{1}{c_j}\right) < 0$, i. e., the weight of subtree T'' decreases while the weight of $T-T''$ remains unchanged. Therefore, there exists an $\epsilon > 0$ such that $\hat{\delta}$ is feasible for $\mathrm{P}(x)$. If $\mathrm{eff}_x(j) < \mathrm{eff}_x(k)$ we would get a contradiction to the optimality of δ. Therefore, $\mathrm{eff}_x(j) = \mathrm{eff}_x(k)$ and we get a solution $\hat{\delta}$ of the 2. Type.
- Assume that there exists a vertex $i \in T - T''$ with $\lambda_i > \delta_i$ and a vertex $j \in T''$ with $\lambda_j < \delta_j$. Then the optimality of λ implies $\mathrm{eff}_x(j) \leq \mathrm{eff}_x(i)$. We can define a new solution where δ_j is decreased and δ_i is increased by some small amount (analogue to the previous discussion). Hence, either we get a contradiction to the optimality of δ or we get a new solution of the 2. Type.
- And finally, assume that there exists a vertex $i \in T - T''$ with $\lambda_i > \delta_i$ and $\lambda_j \geq \delta_j$ for all $j \in T''$. Observe that since $\delta_i < u_i$ and $i \in T - T''$, we know that δ uses the whole budget (because otherwise δ_i could be increased). Moreover, we have

$$\begin{aligned}(w+\lambda)(T-T'') &> (w+\lambda)(T') > (w+\lambda)(T-T') \\ &> (w+\lambda)(T'') \geq (w+\delta)(T'') = (w+\delta)(T-T'').\end{aligned}$$

On the other hand, we have

$$\sum_{j\in T-T''} c_j\lambda_j = B - \sum_{i\in T''} c_i\lambda_i \le B - \sum_{i\in T''} c_i\delta_i = \sum_{j\in T-T''} c_j\delta_j.$$

We use again Lemma 1 to show that there are two vertices $j,k \in T - T''$ with $\lambda_j > \delta_j$ and $\lambda_k < \delta_k$ and $c_j < c_k$. Due to the optimality of λ we have $\mathrm{eff}_x(j) \ge \mathrm{eff}_x(k)$. Now define a new solution $\hat{\delta}$ which is based on δ but δ_j is increased by $\frac{\varepsilon}{c_j}$ and δ_k is decreased by $\frac{\varepsilon}{c_k}$. Obviously, there exists a value $\varepsilon > 0$ such that $\hat{\delta}$ is feasible for $\mathrm{P}'(x)$. Moreover, the weight in $T - T''$ is increased by $\varepsilon\left(\frac{1}{c_j} - \frac{1}{c_k}\right) > 0$ while the weight of T'' remains unchanged. Hence, we get a solution of the 2. Type.

The previous discussion implies that there exists a solution of 2. Type. Let δ be a solution of the second type which minimizes $(w+\delta)(T-T')-(w+\delta)(T') > 0$. This condition is called gap minimality.

- If $\lambda_i \le \delta_i$ for all $i \in T'$ then

  $$(w+\lambda)(T-T') < (w+\lambda)(T') \le (w+\delta)(T') < (w+\delta)(T-T')$$

  and $\sum_{i\in T-T'} c_i\lambda_i \ge \sum_{i\in T-T'} c_i\delta_i$. Then there are two vertices $j,k \in T-T'$ with $\lambda_j < \delta_j$ and $\lambda_k > \delta_k$ and $c_j < c_k$. The optimality of λ implies $\mathrm{eff}_x(j) \le \mathrm{eff}_x(k)$. Then δ can be modified in such a way that δ_j is decreased, δ_k is increased and the weight of $T-T'$ changes by $\varepsilon\left(\frac{1}{c_k} - \frac{1}{c_j}\right) < 0$ for some $\varepsilon > 0$ while the weight of T' remains unchanged. This leads to a contradiction to the gap minimality of δ.
- If there exists a vertex $i \in T'$ with $\lambda_i > \delta_i$ and a vertex $j \in T - T'$ with $\lambda_j < \delta_j$ then $\mathrm{eff}_x(i) \ge \mathrm{eff}_x(j)$ and δ_i can be increased while δ_j can be decreased. Again we get a contradiction to the gap minimality of δ.
- And finally, if there exists a vertex $i \in T'$ with $\lambda_i > \delta_i$ and $\lambda_j \ge \delta_j$ for all $j \in T - T'$. Then

  $$(w+\delta)(T') < (w+\delta)(T-T') \le (w+\lambda)(T-T') < (w+\lambda)(T')$$

  holds. Since $\delta_i < \lambda_i \le u_i$ and $i \in T'$, we know that δ uses the whole budget (otherwise δ_i could be increased). Therefore,

  $$\sum_{i\in T'} c_i\lambda_i = B - \sum_{j\in T-T'} c_j\lambda_j \le B - \sum_{j\in T-T'} c_j\delta_j = \sum_{i\in T'} c_i\delta_i.$$

  Hence, we have two vertices $j,k \in T'$ with $\delta_j < \lambda_j$ and $\delta_k > \lambda_k$ and $c_j < c_k$. The optimality of λ implies $\mathrm{eff}_x(j) \ge \mathrm{eff}_x(k)$. Then δ_j can be increased and δ_k can be decreased. The weight of T' increases by $\varepsilon\left(\frac{1}{c_j} - \frac{1}{c_k}\right) > 0$ while the weight of $T - T'$ remains unchanged. This leads to a contradiction to the gap minimality of δ.

The two previous lemmata are now used to compare optimal solutions of P(x) and P$'(y)$ for $(x,y) \in E$. Given an optimal solution δ of P(x) we have to distinguish whether constraint (3) for $T' \in \mathcal{T}(x)$ with $y \in T'$ is satisfied with equality or strict inequality. We start our investigations with the case of equality in constraint (3):

Lemma 4. *Let δ be an optimal solution of $P(x)$, $(x,y) \in E$ and $y \in T'$ with $T' \in \mathcal{T}(x)$ such that*

$$(w+\delta)(T') = \frac{1}{2}(w+\delta)(T)$$

holds. Then $g(x) < g(y)$ or δ is optimal for $P(y)$ and $g(x) = g(y)$.

Proof. Since subtree T' has exactly half of the total weight of T with respect to $w+\delta$ we conclude that δ is feasible for P(x) and for P(y). The objective value of P(y) is then

$$\begin{aligned}
\sum_{v\in V}(w_v+\delta_v)d(v,y) &= \sum_{v\in T'}(w_v+\delta_v)d(v,y) + \sum_{v\in T-\setminus T'}(w_v+\delta_v)d(v,y) \\
&= \sum_{v\in T'}(w_v+\delta_v)(d(v,x)-\ell(x,y)) \\
&\quad + \sum_{v\in T\setminus T'}(w_v+\delta_v)(d(v,x)+\ell(x,y)) \\
&= g(x) + \ell(x,y)\left((w+\delta)(T-\setminus T') - (w+\delta)(T')\right) \\
&= g(x) + \ell(x,y)\left((w+\delta)(T) - 2(w+\delta)(T')\right) = g(x)
\end{aligned}$$

Hence, δ is feasible for P(y) with objective value $g(x)$. If δ is optimal for P(y) then $g(x) = g(y)$ otherwise $g(x) < g(y)$.

Now we are interested in a neighbour of $x \in V$ such that constraint (3) is met with strict inequality.

Lemma 5. *Let δ be an optimal solution of $P(x)$ such that*

$$(w+\delta)(T') < \frac{1}{2}(w+\delta)(T)$$

for all $T' \in \mathcal{T}(x)$. Then $g(x) \geq g(y)$ holds for all $y \in V$ with $(x,y) \in E$.

Proof. Let $(x,y) \in E$ and let $T' \in \mathcal{T}(x)$ with $y \in T'$. Moreover, $T'' = T - T'$. Since all weight constraints are met with strict inequality, δ is an efficient solution for x. According to Lemma 2 there exists an optimal solution $\tilde{\lambda}$ of P$'(y)$ with $(w+\tilde{\lambda})(T') \leq (w+\tilde{\lambda})(T'')$. Then Lemma 3 implies that there exists an optimal solution $\tilde{\delta}$ of P(y) with $(w+\tilde{\delta})(T') = (w+\tilde{\delta})(T'')$. And finally, Lemma 4 implies $g(x) \geq g(y)$.

Lemma 6. *Let $(x,y) \in E$ and $T' \in \mathcal{T}(x)$ with $y \in T'$. If we have*

$$(w+\delta)(T') < \frac{1}{2}(w+\delta)(T)$$

for every optimal solution δ of $P(x)$ then $g(x) > g(y)$.

Proof. Lemma 3 implies that $(w+\lambda)(T') < (w+\lambda)(T'')$ for all optimal solutions λ of $\mathrm{P}'(x)$. Since λ is an efficient solution for x, Lemma 2 implies that there exists an optimal solution $\tilde{\lambda}$ of $\mathrm{P}'(y)$ with $(w+\tilde{\lambda})(T') \leq (w+\tilde{\lambda})(T'')$. According to Lemma 3 there exists an optimal solution $\tilde{\delta}$ of $\mathrm{P}(y)$ with $(w+\tilde{\delta})(T') = (w+\tilde{\delta})(T'')$. And finally, Lemma 4 implies that either $g(x) > g(y)$ or there exists an optimal solution with equality which is forbidden by assumption.

We are now in the position to prove the following concavity property:

Lemma 7. *Let $(x,y),(y,z) \in E$ with $x \neq z$. If $g(x) \geq g(y)$ then we have $g(y) \geq g(z)$.*

Proof. Let $T_x, T_z \in \mathcal{T}(y)$ with $x \in T_x$, $z \in T_z$ and $T_y = T - (T_x + T_z)$. The situation is pictured in Figure 5.

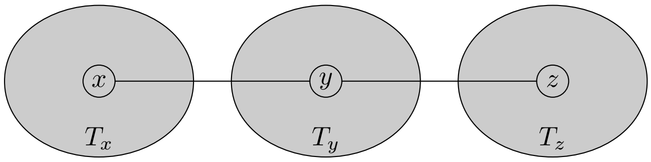

Fig. 5. Illustration for the proof of Lemma 7.

If there exists an optimal solution δ of $\mathrm{P}(y)$ with $(w+\delta)(T') < \frac{1}{2}(w+\delta)(T)$ for all $T' \in \mathcal{T}(y)$ then $g(y) \geq g(z)$ holds (Lemma 5). And if for every optimal solution δ of $\mathrm{P}(y)$ we have $(w+\delta)(T_x+T_y) > (w+\delta)(T_z)$ then $g(y) > g(z)$ (Lemma 6).

Hence, it remains to show that the lemma is true even if for every optimal solution of $\mathrm{P}(y)$ there is one constraint of type (3) which is met with equality and in addition there exists an optimal solution δ of $\mathrm{P}'(y)$ with $(w+\delta)(T_x+T_y) = (w+\delta)(T_z)$.

Observe that there exists an optimal solution $\hat{\delta}$ of $\mathrm{P}'(y)$ with $(w+\hat{\delta})(T_x) = (w+\hat{\delta})(T_y+T_z)$ (because otherwise Lemma 6 would lead to a contradiction to $g(x) \geq g(y)$).

Our goal is to construct a new optimal solution such that no constraint of type (3) is met with equality (which leads to a contradiction): Let $\xi_i = \frac{\delta_i + \hat{\delta}_i}{2}$ (for $i \in V$). Clearly, ξ satisfies the bound and budget constraints. Moreover,

$$\begin{aligned}
(w+\xi)(T_x) &= \frac{1}{2}(w+\delta)(T_x) + \frac{1}{2}(w+\hat{\delta})(T_x) \\
&< \frac{1}{2}(w+\delta)(T_z) + \frac{1}{2}(w+\hat{\delta})(T_y+T_z) \\
&< \frac{1}{2}(w+\delta)(T_y+T_z) + \frac{1}{2}(w+\hat{\delta})(T_y+T_z) \\
&= (w+\xi)(T_y+T_z)
\end{aligned}$$

In an analogue way we can show that $(w+\xi)(T_z) < (w+\xi)(T_x+T_y)$.

Finally, we have to consider subtrees $\tilde{T} \in \mathcal{T}(y)$ with $\tilde{T} \notin \{T_x, T_z\}$:

$$\begin{aligned}(w+\xi)(\tilde{T}) &< (w+\xi)(T_y)\\ &= \frac{1}{2}(w+\delta)(T_y) + \frac{1}{2}(w+\hat{\delta})(T_y)\\ &\le \frac{1}{2}(w+\delta)(T_z) + \frac{1}{2}(w+\hat{\delta})(T_x)\\ &\le \frac{1}{2}(w+\delta)(T_x+T_z) + \frac{1}{2}(w+\hat{\delta})(T_x+T_z)\\ &= (w+\xi)(T_x+T_z) < (w+\xi)(T-\tilde{T})\end{aligned}$$

We have shown that ξ is an optimal solution of $\mathrm{P}(y)$ with

$$(w+\xi)(\tilde{T}) < \frac{1}{2}(w+\xi)(T)$$

for all $\tilde{T} \in \mathcal{T}(y)$ which contradicts the assumption that no such optimal solution exists.

Lemma 7 implies that a local optimum among the function values $g(x)$ for $x \in V$ is also globally optimal. Moreover, we get the following main theorem:

Theorem 4. *Let $T = (V, E)$ be a tree with vertex weights $w_v \in \mathbb{R}_+$, cost coefficients $c_v \in \mathbb{R}_+$ and bounds $u_v \in \mathbb{R}_+$ for $v \in V$, edge lengths $\ell_e \in \mathbb{R}_+$ for $e \in E$ and budget B. Let $x \in V$ and let λ be an optimal solution of $P'(x)$.*

- *If $(w+\lambda)(T') < \frac{1}{2}(w+\lambda)(T)$ for all $T' \in \mathcal{T}(x)$ then $g(x)$ is the optimal objective value of Down1Median and vertex x is the* 1*-median with respect to an optimal weight modification.*
- *If there exists a subtree $T' \in \mathcal{T}(x)$ with $(w+\lambda)(T') \ge \frac{1}{2}(w+\lambda)(T)$ then there exists a* 1*-median $x^* \in V$ with respect to optimal weight modifications such that $x^* \in T'$.*

Proof. If $(w+\lambda)(T') < \frac{1}{2}(w+\lambda)(T)$ for all $T' \in \mathcal{T}(x)$ then λ is an optimal solution of $P(x)$. According to Lemma 5 we have $g(x) \ge g(y)$ for all $y \in \Gamma(x)$. The result follows together with Lemma 7.

If there exists a subtree $T' \in \mathcal{T}(x)$ with $(w+\lambda)(T') \ge \frac{1}{2}(w+\lambda)(T)$ then there exists an optimal solution δ of $\mathrm{P}(x)$ with $(w+\lambda)(T') = \frac{1}{2}(w+\lambda)(T)$ (Lemma (3)). According to Lemma 4 we know that $g(x) \le g(y)$ for $(x, y) \in E$ and $y \in T'$. Then Lemma 7 implies $g(x) \ge g(z)$ for all $z \in \Gamma(x) \setminus \{y\}$ and herewith for all $z \in T - (\{x\} \cup T')$. Hence, there exists an optimal solution in T'.

Observe that Theorem 4 immediately implies the following algorithm for Down1Median on trees:

- Step 1 (Initialization): Let $\tilde{T} = (\tilde{V}, \tilde{E}) = (V, E)$.

– Step 2 (Binary search): If $\tilde{V} = \{x, y\}$ and both vertices are already investigated then go to Step 5. Else, let $x \in \tilde{V}$ be a centroid of $\tilde{T}$ that was not already investigated, i. e., x is a vertex such that $|V'| \leq \frac{1}{2}|\tilde{V}|$ for all subtrees T' in $\tilde{T}$ that arise if x is deleted.
– Step 3 (Investigation of x): Solve $\mathrm{P}'(x)$. Let λ be an optimal solution of $\mathrm{P}'(x)$. If $(w+\lambda)(T') < \frac{1}{2}(w+\lambda)(T)$ for all $T' \in \mathcal{T}(x)$ then x is an optimal vertex and λ is an optimal solution for Down1Median in T. If $\tilde{V} = \{x\}$ then go to Step 5, otherwise go to Step 4.
– Step 4 (Reduction of search space): Let $T' = (V', E') \in \mathcal{T}(x)$ such that $(w+\lambda)(T') \geq \frac{1}{2}(w+\lambda)(T)$. Set $\tilde{V} = V'$ and $\tilde{E} = E'$ with $y \in V'$. Go to Step 2.
– Step 5 (Determine solution for optimal vertex): An optimal solution of $\mathrm{P}(x)$ is optimal for Down1Median on T.

Theorem 4 implies the correctness of this algorithm. Observe that a centroid can be found in linear time (Handler and Mirchandani [17]). Moreover, since we delete approximately half of the vertices in every iteration, we have $\mathcal{O}(\log n)$ iterations. In fact we do a binary search on the tree. Instead of solving $\mathrm{P}(x)$ to optimality in every iteration we solve only the relaxed version $\mathrm{P}'(x)$ which still gives enough information on an optimal solution of $\mathrm{P}(x)$.

If we get an optimal solution λ of $\mathrm{P}'(x)$ which is also feasible for $\mathrm{P}(x)$ and satisfies all constraints of type (3) with strict inequality, then λ is an optimal solution of Down1Median. However, if we end up with two vertices $x, y \in V$ with

$$(w+\lambda^x)(T') \geq (w+\lambda^x)(T'')$$
$$(w+\lambda^y)(T') \leq (w+\lambda^y)(T'')$$

where λ^x (λ^y) is an optimal solution of $\mathrm{P}'(x)$ ($\mathrm{P}'(y)$) and $T' \in \mathcal{T}(x)$ with $y \in T'$ and $T'' \in \mathcal{T}(y)$ with $x \in T''$ then Lemmata 3 and 4 imply $g(x) = g(y)$. Together with Lemma 7 we know that $g(x) = g(y)$ is the optimal objective value of Down1Median of T. If λ^x is not feasible for $\mathrm{P}(x)$ then there exists a subtree $T' \in \mathcal{T}(x)$ whose weight is strictly greater than one half of the total weight. Hence, there exists an optimal solution δ of $\mathrm{P}(x)$ such that the weight of T' is exactly half of the total weight. Let δ be an optimal solution of $\mathrm{P}'(x)$ with the additional constraint that $(w+\delta)(T') = \frac{1}{2}(w+\delta)(T)$ then δ is an optimal solution of $\mathrm{P}(x)$. Observe that this new problem has two constraints in addition to the upper bound constraints and can therefore be solved in linear time (Megiddo and Tamir [20]).

Theorem 5. *Downgrading the* 1*-median on a tree with vertex weight modifications can be solved in* $\mathcal{O}(n \log n)$ *time.*

Proof. The correctness of the above algorithm was already shown before. It remains to analyze the running time. There are $\mathcal{O}(\log n)$ iterations and in each iteration we have to find a centroid and an optimal solution of a continuous knapsack problem. Both subproblems can be solved in linear time. Hence, an optimal vertex can be found in $\mathcal{O}(n \log n)$ time. As soon as an optimal vertex is found the corresponding optimal vertex weight modification has to be determined in Step 5. Solve $\mathrm{P}'(x)$ together with one weight constraint in order to get an optimal solution. This last linear programming problem can be solved in linear time. Therefore, the whole algorithm takes $\mathcal{O}(n \log n)$ time.

5 Conclusion

This paper deals with changing parameters of a network such that the optimal objective value of the 1-median problem in the modified network is minimized (upgrading) and maximized (downgrading), respectively. In many cases upgrading problems can be lead back to reverse problems. In contrast, downgrading problems have a min-max-structure and therefore need a different approach. In case of the 1-median problem the downgrading version can be written als linear programming problem and for the special case of a tree an $\mathcal{O}(n \log n)$ time algorithm is developed. It seems to be interesting to apply up- and downgrading versions to different underlying location problems like p-median or p-center problems. Another direction of future research is the investigation of up- and downgrading problems with variable edge lengths.

References

1. E. Balas and E. Zemel, An algorithm for large zero-one knapsack problems, *Operations Research* 28, 1130-1154, 1980.
2. R. E. Burkard, E. Gassner, and J. Hatzl, A linear time algorithm for the reverse 1-median problem on a cycle, *Networks* 48, No. 1, 16-23, 2006.
3. R. E. Burkard, E. Gassner, and J. Hatzl, The reverse 2-median problem on trees, accepted for publication in *Discrete applied mathematics*.
4. R.E. Burkard, B. Klinz, and J. Zhang, Bottleneck capacity expansion problems with general budget constraints, *RAIRO Recherche Operationnelle* 35, 1-20, 2001.
5. R.E. Burkard, Y. Lin, and J. Zhang, Weight reduction problems with certain bottleneck objectives, *European Journal of Operational Research* 153, 191-199, 2004.
6. R.E. Burkard, C. Pleschiutschnig, and J. Zhang, Inverse median problems, *Discrete Optimization* 1, 23-39, 2004.
7. M.C. Cai, X.G. Yang, and J. Zhang, The complexity analysis of the inverse center location problem, *Journal of Global Optimization* 15, No. 2, 213-218, 1999.
8. K.U. Dragmeister, S.O. Krumke, M.V. Marathe, H. Noltemeier, and S.S. Ravi, Modifying edges of a network to obtain short subgraphs, *Theoretical Computer Science* 203, 91-121, 1998.
9. Z. Drezner and H. W. Hamacher, *Facility location. Applications and theory.* Springer, Berlin, 2002.
10. G.N. Frederickson and R. Solis-Oba, Increasing the weight of minimum spanning trees, *Journal of Algorithms* 33, No. 2, 244-266, 1999.
11. D.R. Fulkerson and G.C. Harding, Maximizing the minimum source-sink path subject to a budget constraint, *Mathematical Programming* 13, 116-118, 1977.
12. E. Gassner, The inverse 1-maxian problem on a tree, accepted for publication in *Journal of Combinatorial Optimization*.

13. E. Gassner, Up- and downgrading the 1-center in a network, submitted.
14. A.J. Goldman, Optimal center location in simple networks, *Transportation Science* 5, 212-221, 1971.
15. S.L. Hakimi, Optimal Location of switching centers and the absolute centers and medians of a graph, *Operations Research* 12, 450-459, 1964.
16. S. E. Hambrusch and Hung-Yi Tu, Edge Weight Reduction Problems in Directed Acyclic Graphs, *Journal of Algorithms* 24, No. 1, 66-93, 1997.
17. G.Y. Handler and P.B. Mirchandani, *Location on networks - Theory and Algorithms*, MIT Press, Cambridge, MA, 1979.
18. O. Kariv and S. L. Hakimi, An algorithmic approach to network location problems. I: The p-centers and II: The p-medians, *SIAM Journal of Applied Mathematics* 37, No. 3, 513-560, 1979.
19. S.O. Krumke, M.V. Marathe, H. Noltemeier, R. Ravi, and S.S. Ravi, Approximation algorithms for certain network improvement problems, *Journal of Combinatorial Optimization* 2, 257-288, 1998.
20. N. Megiddo and A. Tamir, Linear time algorithms for some separable quadratic programming problems, *Operations Research Letters* 13, No. 4, 203-211, 1993.
21. P. B. Mirchandani and R. L. Francis, *Discrete location theory*. Interscience Series in Discrete Mathematics and Optimization. New York etc.: John Wiley & Sons Ltd., 1990.
22. C. A. Phillips, The network inhibition problem, *Annual ACM Symposium on Theory of Computing*, Proceedings of the twenty-fifth annual ACM symposium on Theory of computing table of contents San Diego, California, United States, 776-785, 1993.
23. J. Zhang, X.G. Yang and M.C. Cai, Inapproximability and a polynomially solvable special case of a network improvement problem, *European Journal of Operational Research* 155, 251-257, 2004.
24. J. Zhang, X.G. Yang and M.C. Cai, A network improvement problem under different norms, *Computational Optimization and Applications* 27, No. 3, 305-319, 2004.